

### Q1. What is the difference between these two snippets?

```
$('#button').on('click', function () {  
    alert('you clicked the button!');  
});  
$('#button').click(function () {  
    alert('you clicked the button!');  
});
```

- ☐ Only the second one will work; jQuery does not have a function called `.on`.
- ☐ The second snippet will not function.
- ☒ Nothing `.click(function)` is shorter way to write `.on('click', function)`.
- ☐ The first snippet will execute for every button on the page, the second will only apply to the first button.

### Q2. What does the following line of code do?

```
jQuery('p')
```

- ☐ Loads a paragraph tag from a remote server using AJAX
- ☐ Aliases jQuery to a variable `p`
- ☒ Selects all paragraphs on the page
- ☐ Creates a new paragraph tag and inserts it into the body tag

### Q3. Given the following HTML, how could we use one line to hide or show the button?

```
<button class="btn btn-primary" type="submit">Continue to checkout</button>
```

- ☒ `$('.btn-primary').toggle();`
- ☐ `$('.btn-primary').showHide();`
- ☐ `$('.btn-primary').not(':visible').show();`
- ☐ `$('.btn-primary').css({ display: 'block' });`

### Q4. Working with AJAX, we may run into situations where a piece of code should not be run until after multiple AJAX calls have completed successfully. Say we need to call two external services for JSON data (a list of students, and a list of classes). And only after retrieving those data will we perform some manipulations on a page. What is the preferred way for dealing with this scenario?

```
https://example.com/json-api/students  
https://example.com/json-api/classes
```

- ☐ A

```
$.get(
  ['https://example.com/json-api/students', 'https://example.com/json-
  api/classes'],
  function (studentRequest, classRequest) {
    // the rest of the code goes here
  },
);
```

- ☐ B

```
$.when(
  $.get('https://example.com/json-api/students'),
  $.get('https://example.com/json-api/classes'),
).done(function (studentRequest, classRequest) {
  // the rest of the code goes here
});
```

- ☐ C

```
$.bind(
  $.get('https://example.com/json-api/students'),
  $.get('https://example.com/json-api/classes'),
).done(function (studentRequest, classRequest) {
  // the rest of the code goes here
});
```

- ☒ D

```
$.ajax('https://example.com/json-api/students', {
  success: function (studentRequest) {
    $.ajax('https://example.com/json-api/classes', {
      success: function (classRequest) {
        // the rest of the code goes here
      },
    });
  },
});
```

**Q5. Given the snippet of HTML below, what is the difference between the two lines that follow it?**

```
<ul>
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
  <li>Item 4</li>
</ul>

$('ul').find('li').get(2);
$('ul').find('li').eq(2);
```

- ☒ .get() retrieves a DOM element, and can't be chained, eq() retrieves a jQuery object, and can be chained.
- ☐ .get() retrieves a jQuery object, and can't be chained, eq() retrieves a DOM element, and can be chained.
- ☐ .get() retrieves a jQuery object, and is zero-indexed, eq() retrieves a DOM element, and is 1-indexed.
- ☐ .get() retrieves a DOM element, and is zero-indexed, eq() retrieves a jQuery object, and is 1-indexed.

**Q6. Suppose we want to have a ball created from an HTML element (id=ball) move down and to the right from its original location when clicked, and move back into its original place when finished. Given a starting point of this, which of these snippets would accomplish that goal?**

```
$('#ball').click(function () {
  // Our code goes here
});
```

- ☒ A

```
$(this).animate(
  { top: '+=100', left: '+=100' },
  {
    duration: 600,
    complete: function () {
      $(this).animate({ top: '-=100', left: '-=100' }, 600);
    },
  },
);
```

- ☐ B

```
$(this).animate({ top: '-=100', left: '-=100' }, 600, function () {
  $(this).animate({ top: '+=100', left: '+=100' }, 600);
});
```

- ☐ C

```
$(this).animate(
  { top: '=100', left: '=100' },
  {
    duration: 600,
    complete: function () {
      $(this).animate({ top: 0, left: 0 }, 600);
    },
  },
);
```

- ☐ D

```
$(this).animate({ top: '100', left: '100' }, 600, function () {  
    $(this).animate({ top: 0, left: 0 }, 600);  
});
```

**Q7. Given the following CSS and HTML codes below, how could you apply the success class to the feedback div?**

```
.success {  
    color: green;  
    background: #ddffdd;  
}
```

```
<div class="feedback">Thank you for answering this survey.</div>
```

- ☐ `$('.feedback').hasClass('.success');`
- ☐ `$.css('.feedback', '.success');`
- ☒ `$('.feedback').addClass('success');`
- ☐ `$('.feedback').css('.success');`

**Q8. The following page snippet includes a couple of messages in a list, and a code snippet that retrieves a few hundred messages from an API endpoint using AJAX. How can you add these new items to the .message-area--list element in the most performant way?**

```
<div class="message-area">  
    <ul class="message-area--list">  
        <li>Existing message 1</li>  
        <li>Existing message 2</li>  
    </ul>  
</div>
```

```
$.get('//example.com/api/v1/message').done(function (data) { var tonsOfItems =  
data.messages; // add  
all these messages to a large page });
```

- ☐ A

```
tonsOfItems.map(function (item) {  
    $('.message-area--list').append('<li>' + item + '</li>');  
});
```

- ☒ B

```
var tonsOfListItems = tonsOfItems.map(function (item) {  
    return '<li>' + item + '</li>';  
});  
$('.message-area--list').append(tonsOfListItems.join(''));
```

- ☐ C

```
CSS.$messageList = $('.message-area--list');
```

```
$.each(tonsOfItems, function (idx, item) {  
    $('<li>' + item + '</li>').appendTo($messageList);  
});
```

- ☐ D

```
$.each(tonsOfItems, function (idx, item) {  
    $('.message-area--list').append('<li>' + item + '</li>');  
});
```

## Q9. What is jQuery?

- ☐ jQuery is a bridge between Java and Javascript that makes native apps easier to write.
- ☐ jQuery is a plugin for JavaScript that makes database queries easier to write.
- ☒ jQuery is a collection of JavaScript functions that makes finding and manipulating elements on a page, AJAX, and other things easier.
- ☐ jQuery is a Chrome extension that allows users to create their own extensions with just a few lines of JavaScript.

## Q10. We want to create a new jQuery plugin called linkUpdater that can be chained onto other jQuery selector like a normal plugin. It should update all the links in the referenced collection so they open in new windows or tabs. Below is the first cut. What is one problem with this plugin?

```
'user strict';  
($.linkUpdater = function () {  
    this.find('a').attr('target', '_blank');  
})(jQuery);
```

- ☐ this needs to be wrapped, like \$(this), in order to be chained in a plugin.
- ☐ jQuery plugins can't be safely authored in strict mode.
- ☐ In order to be used by other code, plugins need to be added to the global namespace, not wrapped in a function expression.
- ☒ Our plugin should extend jQuery.fn, not jQuery itself.

## Q11. Generally speaking, when used on a web page, how should jQuery be installed, and why?

- ☒ Just before the closing body tag, because we want to avoid blocking other resources from loading, and we use the ready method to make sure our code fires after the DOM is ready

- ☐ Using the highest version number possible because only jQuery 3 and up are compatible with Internet Explorer 7
- ☐ In the head tag because we want jQuery available as soon as possible
- ☐ From a CDN because we want to be able to use jQuery online or offline

**Q12. Given the following HTML, how could we make this button disappear from the page using jQuery?**

```
<button class="btn btn-primary" type="submit">Continue to checkout</button>
```

- ☒ `$('.btn-primary').hide();`
- ☐ `$('.btn-primary:visible').not();`
- ☐ `$('.btn-primary').visibility(false);`
- ☐ `$('.btn-primary').show(false);`

**Q13. What is the difference between `$('#header').html()` and `$('#header').text()`?**

- ☒ `$('#header').html()` returns the inner HTML of the header. `$('#header').text()` returns only the text
- ☐ `$('#header').html()` returns only the HTML tags used, without the text. `$('#header').text()` returns only the text
- ☐ `$('#header').html()` strips all HTML from the header. `$('#header').text()` always returns an empty string.
- ☐ `$('#header').html()` returns all headers in an HTML document. `$('#header').text()` the first line of a text file.

**Q14. When writing jQuery plugins, we often provide default options that may be overridden by the end user. What jQuery function is most useful for this purpose?**

- ☒ `$.extend`
- ☐ `$.clone`
- ☐ `$.fn.extend`
- ☐ `$.merge`

**Q15. There are times when you might want to programmatically trigger an event, instead of simply reacting to user input directly. Given this markup, Which choice will NOT cause a click event to the select box when the button is clicked?**

```
<article>
```

```
<div>Here's a button you can click: <button class="btn">Click Me</button></div>
<form>
  <p>Further down the page, there's a select box.</p>
  <select>
    <option value="1">One</option>
    <option value="2">One</option>
    <option value="3">One</option>
    <option value="4">One</option>
  </select>
</form>
</article>
```

- ☒ `$('#button').on('click.myApp', (function() {  
 $('#input[type=select]').trigger('click'); }));`
- ☐ `$('#button').on('click', (function() { $('#input[type=select]').click(); }));`
- ☐ `$('#button').trigger(function() { $('#input[type=select]').click(); });`
- ☐ `$('#button').click(function() { $('#input[type=select]').click(); });`

**Q16. You have an absolutely positioned element inside a relatively positioned parent element, and you want to animate that element within its parent element. What jQuery function is most useful for finding the initial coordinates of the .animate-me?**

```
<style>
  .parent {
    position: relative;
    top: 3em;
    width: 50%;
    min-height: 50vh;
    margin: 0 auto;
  }

  .animate-me {
    position: absolute;
    top: 40px;
    right: 30px;
  }
</style>

<div class="parent">
  <div class="animate-me">This box will move!</div>
</div>
```

- ☒ `$('.animate-me').offset();`
- ☐ `$('.animate-me').each();`
- ☐ `$('.animate-me').position();`
- ☐ `$('.animate-me').offsetParent();`

**Q17. You want to work with AJAX using a Promise-like interface instead of nested callback functions. What jQuery API should you use?**

- ☐ jQuery.sub
- ☐ jQuery.ajaxTransport
- ☒ jQuery.Deferred
- ☐ jQuery.proxy

**Q18. What is tricky about jQuery's nth- filters (:nth-child, :nth-of-type, etc.) relative to other filters?**

- ☒ Referring to lists of items, they are 1-indexed (like CSS), not 0-indexed (like JavaScript).
- ☐ They don't return the jQuery object, and cannot be chained.
- ☐ They can return the wrong items if the DOM was recently manipulated.
- ☐ They are not part of CSS, so they don't get the performance benefits of passing through the `document.querySelector`.

**Q19. jQuery's AJAX functions return objects that implement the Promise API. As a result, you can chain promises and avoid nested callbacks. What does that look like?**

- ☒ A

```
$.get('http://httpbin.org/delay/2')
  .then(function (response) {
    // Data from first GET is here as 'response'
    return $.get('http://httpbin.org/delay/2');
  })
  .then(function (response) {
    // Data from second GET is here as 'response'
  });
```

- ☐ B

```
$.get('http://httpbin.org/delay/2')
  .catch(function (response) {
    // Data from first GET is here as 'response'
    return $.get('http://httpbin.org/delay/2');
  })
  .done(function (response) {
    // Data from second GET is here as 'response'
  });
```

- ☐ C

```
$.get('http://httpbin.org/delay/2', function (response1) {
  // Data from first GET is here as 'response1'

  $.get('http://httpbin.org/delay/2', function (response2) {
```



```
    // Data from second GET is here as 'response2'
  });
});
```

- ☐ D

```
$.get('http://httpbin.org/delay/2')
  .then(function (response) {
    // Data from first GET is here as 'response'
    return response;
  })
  .get('http://httpbin.org/delay/2', function (response) {
    // Data from second GET is here as 'response'
  });
```

**Q20. You want to have a ball that is created from an HTML element (id=ball) move down and to the right of its original location when clicked, and move back to its original place when finished. What snippet, added to the code below, would do this?**

```
$('#ball').click(function () {
  // Our code goes here
});
```

- ☐ A

```
$(this).animate(
{
  top: '-=100',
  left: '-=100',
},
600,
function () {
  $(this).animate(
  {
    top: '+=100',
    left: '+=100',
  },
  600,
  );
},
);
```

- ☒ B

```
$(this).animate(
{
  top: '+=100',
  left: '+=100',
},
{
  duration: 600,
  complete: function () {
    $(this).animate(
```

```

        {
            top: '-=100',
            left: '-=100',
        },
        600,
    );
},
},
);

```

•  C

```

$(this).animate(
{
    top: 100,
    left: 100,
},
600,
function () {
    $(this).animate(
        {
            top: 0,
            left: 0,
        },
        600,
    );
},
);

```

•  D

```

$(this).animate(
{
    top: 100,
    left: 100,
},
{
    duration: 600,
    complete: function () {
        $(this).animate(
            {
                top: 0,
                left: 0,
            },
            600,
        );
    },
},
);

```

**Q21. The way `.wrap()` works is sometimes misunderstood. Given the DOM and jQuery snippets below, what does the modified DOM snippet look like?**

```

<div id="container">
  <div class="item">Here's an item</div>
</div>

```

```
$('#container').wrap('<div class="wrapper"></div>').css('border', '2px solid red');
```

- ☐ A

```
<div class="wrapper" style="border: 2px solid red;">
  <div id="container">
    <div class="item">Here's an item</div>
  </div>
</div>
```

- ☒ B

```
<div class="wrapper">
  <div id="container" style="border: 2px solid red;">
    <div class="item">Here's an item</div>
  </div>
</div>
```

- ☐ C

```
<div id="container" style="border: 2px solid red;">
  <div class="wrapper">
    <div class="item">Here's an item</div>
  </div>
</div>
```

- ☐ D

```
<div id="container">
  <div class="wrapper" style="border: 2px solid red;">
    <div class="item">Here's an item</div>
  </div>
</div>
```

**Q22. How can you select the following blockquote AND the list in a single call to jQuery() without chaining?**

```
<div class="quotes">
  <blockquote data-favorite="false">A quote</blockquote>
  <blockquote data-favorite="true">A favorite quote</blockquote>
  <blockquote data-favorite="false">A quote</blockquote>
  <blockquote data-favorite="false">A quote</blockquote>
</div>
```

```
<ul class="menu-first">
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
  <li>Item 4</li>
</ul>
```

- ☐ `$('.quotes + .menu-first')`

- ☐ `$('.quotes .menu-first')`
- ☒ `$('.quotes, .menu-first')`
- ☐ `$('.quotes' + '.menu-first')`

**Q23. Effects like `show`, `hide`, `fadeIn`, and `fadeOut` can be called with no arguments, but can also take arguments for how long they should last. Which is NOT a duration argument supported by these functions?**

- ☐ "fast"
- ☒ "extreme"
- ☐ 2000
- ☐ "slow"

**Q24. Though jQuery offers visual effects, it is considered a best practice to use CSS to set up different states triggered by classes, where it makes sense. What's the easiest way to enable and disable a class bounce on an element with the ID dialog?**

- ☐ `$('#dialog').toggleClass('bounce')`
- ☐ `$('#dialog.bounce').removeClass().addClass()`
- ☐ `$('#dialog').addOrRemoveClass('bounce')`
- ☒ `$('#dialog').toggleClass('bounce')`

**Q25. What is the main difference between selectors and filters?**

- ☐ Selectors are used to refine the content that filters have been applied to.
- ☒ Selectors are used to find and select content in a page. Filters are used to refine the results of selectors.
- ☐ Filters are used to remove content from the page. Selectors are used to add content to the page
- ☐ There is no real difference. They are both used to build up lists of page content.

**Q26. You want to create a custom right-click menu. How might you start the code?**

- ☐ `$('#canvas').on('click.right', function(){ console.log('Handled a right-click') });`
- ☐ `$('#canvas').on('contextual', function(){ console.log('Handled a right-click') });`

- ☒ `$('#canvas').on('contextmenu', function(){ console.log('Handled a right-click') });`
- ☐ `$('#canvas').on('rightclick', function(){ console.log('Handled a right-click') });`

**Q27. What is the correct way to check how many paragraphs exist on a page using jQuery?**

- ☐ `$('p').count()`
- ☒ `$('p').length`
- ☐ `$('#*').find('p')`
- ☐ `$('p').length()`

**Q28. As with many areas of JavaScript, keeping track of the meaning of this is important and sometimes tricky. What does this mean at each of the two points in this custom plugin snippet?**

```
$.fn.customPlugin = function () {
  // Point 1

  return this.each(function () {
    // Point 2
  });
};
$(document).customPlugin();
```

- ☐ At Point 1, this means a jQuery object. At Point 2, it means a DOM element.
- ☐ In this case, they mean the same thing: a jQuery object.
- ☐ In this case, they mean the same thing: a DOM element.
- ☒ At Point 1, this means a DOM element. At Point 2, it means a jQuery object.

**Q29. How can you make the first list item bold and the next item oblique, in a single statement chain?**

```
<ul class="menu-first">
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
  <li>Item 4</li>
</ul>
```

- ☐ A

```
$('.menu-first > li').eq(0).css('font-weight', 'bold').eq(1).css('font-style', 'oblique');
```

- ☐ B

```
$('.menu-first > li').first().css('font-weight', 'bold').after().css('font-style', 'oblique');
```

- ☐ C

```
$('.menu-first > li').first().css('font-weight', 'bold').second().css('font-style', 'oblique');
```

- ☒ D

```
$('.menu-first > li').eq(0).css('font-weight', 'bold').next().css('font-style', 'oblique');
```

### Q30. Which CSS selectors can you NOT use in jQuery?

- ☐ You cannot use multiple class selectors such as .class1.class2.
- ☐ You cannot use pseudo-classes such as :not or :last-of-type.
- ☐ You cannot use IDs and classes together, such as #element.class.
- ☒ None. All CSS selectors are compatible in jQuery.

**Q31. Starting with some DOM elements in the nested structure below, you assign listeners for the same event to a child element and one of the parents using the JavaScript that follows. You want to ensure that when .leaf is clicked, only its event handler will be fired, instead of the click bubbling up and also firing the parent's click handler. What do you need to add to its handler function?**

```
<ul class="items" id="main-menu">
  <li>
    Item 1
    <ul>
      <li class="leaf">Sub Item 1</li>
      <li>Sub Item 2</li>
    </ul>
  </li>
</ul>
```

```
$('.leaf').click(function (event) {
  console.log('Sub Item 1 got a click');
});
$('#main-menu').click(function (event) {
  console.log('Main menu got a click');
});
```

- ☐ event.capture();

- ☒ `event.stopPropagation();`
- ☐ `event.preventDefault();`
- ☐ `event.stop();`

**Q32. Using event delegation, you can listen for events on a lot of different items without having to attach separate listeners to each one. But there are times when you may want to check the type of item receiving the event before doing anything, such as checking if an image was clicked versus a text field. Given the starter code below, which choice shows what jQuery provides to help with that process?**

```
<div id="sidebar">
  
  <input type="text" placeholder="Fill in something" />
</div>

$('#sidebar').click(function (evt) {
  var $target = $(evt.target);

  // What goes here?
});
```

- ☐ `$( $target.get(0) + ':image' )`
- ☐ `$( 'img' ).is( $target )`
- ☐ `$target.filter( 'img' )`
- ☒ `$target.is( 'img' )`

**Q33. There are many ways to create elements that can be added to the page. Which answer is NOT one of those ways, assuming you have the following on the page?**

```
<div id="elements"></div>
```

- ☐ A

```
$('#elements').append($('<p class="appended">As an HTML string</p>'));
```

- ☐ B

```
var p = document.createElement('p');
var text = document.createTextNode('As a DOM element');
p.appendChild(text);
$('#elements').append(p);
```

- ☒ C

```
$('#elements').append(<p class="appended">As a JSX object</p>);
```

- ☐ D

```
$('#elements').append(  
  $('<p>', {  
    class: 'appended',  
    text: 'As an attribute object',  
  }),  
);
```

**Q34. The .addClass() and .removeClass() methods can accept functions as arguments. What does this function do?**

```
$('#menu').addClass(function () {  
  return $('body').attr('class');  
});
```

- ☐ It adds the first class found on the body element to the #menu element.
- ☐ It adds all classes found on the #menu element to the body tag.
- ☐ It replaces any classes on the #menu element with all classes from the body tag.
- ☒ It adds all classes found on the body element to the #menu element.

**Q35. You're working on a site that uses an old version of jQuery, and you want to update to a newer version. What's the most efficient way to do so?**

- ☐ Install the newer version of jQuery, go through each script one by one, and fix what looks broken.
- ☐ Read the change notes for the newer version of jQuery, fix all scripts, install the newer version, and fix anything that remains broken.
- ☒ Install the newer version of jQuery as well as its Migrate plugin, fix all warnings, and uninstall the Migrate plugin.
- ☐ Install the newer version of jQuery at the same time, and use jQuery.noConflict() on pages that need the older version.

**Q36. Let's say you have a page with just one link on it. How can you change the anchor tag so it links to example.com?**

- ☐ \$('a').attribute('href', 'http://www.example.com')
- ☒ \$('a').attr('href', 'http://www.example.com')
- ☐ \$('a').data('href', 'http://www.example.com')
- ☐ \$('a').href('http://www.example.com')

**Q37. What does \$() mean in jQuery?**



- ☒ It is an alias to the main core method of jQuery itself—the same as writing `jQuery()`.
- ☐ It is a utility function that selects the first element from the document.
- ☐ It is a shorter way to write `document.getElementById()`.
- ☐ It is a utility function that selects the last element from the document.

**Q38. Along with DOM traversal and manipulation, jQuery offers several general-purpose helper functions that fill in some JavaScript gaps, especially before ES2015. Which is NOT a jQuery utility function?**

- ☐ `jQuery.each`, a general purpose iterator for looping over arrays or objects
- ☐ `jQuery.isNumeric`, which can check whether its argument is, or looks like, a number
- ☐ `jQuery.extend`, which can merge objects and make complete deep copies of objects
- ☒ `jQuery.isMobile`, which can tell whether the user is using a mobile browser

**Q39. Given this set of checkboxes, how can you select the ones that have the phrase "sun" as part of the value?**

```
<input type="checkbox" name="artists[]" value="sun-ra" />
<input type="checkbox" name="artists[]" value="otis-redding" />
<input type="checkbox" name="artists[]" value="captain-beefheart" />
<input type="checkbox" name="artists[]" value="king-sunny-ade" />
<input type="checkbox" name="artists[]" value="weather-report" />
```

- ☐ `$('checkbox').val(/sun/);`
- ☒ `$('input[value*="sun"]');`
- ☐ `$('input[value|= "sun"]');`
- ☐ `$('input:checkbox').attr('value', '*sun*');`

**Q40. How can you get an AJAX request to go through without triggering any of jQuery's AJAX events?**

- ☐ Set the `type` option to "none".
- ☐ Set the `processData` option to false.
- ☐ Set a success callback that returns false.
- ☒ Set the option "global" to false.

**Q41. How do you change the current value of a text field with the class `.form-item` to "555-1212"?**

- ☐ `$.val('.form-item', '555-1212');`
- ☒ `$('.form-item').val('555-1212');`
- ☐ `$('.form-item').data('value', '555-1212');`
- ☐ `$('.form-item').set('value', '555-1212');`

**Q42. How would you fire a callback when any AJAX request on a page has completed?**

- ☐ `$('#body').ajaxComplete(function() { console.count('An AJAX request completed'); });`
- ☐ `$(document).on('ajax-complete', function() { console.count('An AJAX request completed'); });`
- ☐ `$('#body').on('ajaxComplete', function() { console.count('An AJAX request completed'); });`
- ☒ `$(document).ajaxComplete(function() { console.count('An AJAX request completed'); });`

**Q43. Given this set of checkboxes, how can you select the one with the value "blimp"?**

```
<input type="checkbox" name="songs[]" value="satisfaction" />
<input type="checkbox" name="songs[]" value="respect" />
<input type="checkbox" name="songs[]" value="blimp" />
<input type="checkbox" name="songs[]" value="saturn" />
<input type="checkbox" name="songs[]" value="penguins" />
```

- ☒ `$('#input[value="blimp"]');`
- ☐ `$('#input[value!="blimp"]');`
- ☐ `$('#checkbox').val('blimp');`
- ☐ `$('#input:checkbox').attr('value', 'blimp');`

**Q44. Given this snippet of HTML and jQuery code, what does the jQuery do?**

```
<ul class="menu">
  <li><a href="#" class="active">Home</a></li>
  <li><a href="#">Page 2</a></li>
</ul>
<ul class="active submenu">
  <li><a href="#">Subpage 1</a></li>
  <li><a href="#">Subpage 2</a></li>
</ul>

var m = $('.menu'),
    sm = $('.submenu');
m.add(sm);
m.css('font-weight', 'bold');
```

- ☐ It makes all the menu items bold.

- ☐ It throws an exception on line 3.
- ☒ It makes the first set of menu items, not the second, bold.
- ☐ It makes the second set of menu items, not the first, bold.

**Q45. You want to take a block of type and animate it to a larger size with jQuery. The following HTML and JavaScript behaves strangely. What is the issue?**

```
<div id="type" style="font: 1em/1.5 helvetica, arial, sans-serif; background:
#ffc; padding: 0">
  Animate me!
</div>

$('#type').animate(
{
  fontSize: '+=1em',
},
3000,
);
```

- ☐ jQuery does not support ems and will make the type 1 pixel larger instead of 1 em larger.
- ☐ jQuery cannot override CSS in a style attribute, so the font size will not change.
- ☐ The font size was set with a shorthand property, so jQuery will not animate the font size at all.
- ☒ The font size was set with a shorthand property, so jQuery will start the animation from 0 instead of from 1 em.

**Q46. When using the `clone()` function to duplicate an element, what is one of the main concerns your code needs to watch out for?**

- ☐ The `clone()` function may ignore data attributes on the original elements.
- ☒ The `clone()` function may result in elements with duplicate ID attributes.
- ☐ The `clone()` function may remove CSS classes from the cloned elements.
- ☐ The `clone()` function may not respect the attribute order of the original elements.

**Q47. When incorporating a plugin into a project, what are the important steps for basic installation and usage?**

- ☒ The jQuery script tag must come first, followed by the plugin, followed by your custom scripts, all preferably at or near the bottom of the page.

- ☐ Your custom scripts must appear first in the document <head>, followed by jQuery, followed by the plugin.
- ☐ The jQuery script tag and the plugin script tag must appear in the document <head>, and your custom scripts can follow anywhere on the page.
- ☐ The jQuery script tag must appear in the document <head>, but the plugin and your custom scripts can appear anywhere else in any order.

**Q48. These two script tags show different ways of using jQuery's ready() method. What is true about both approaches?**

```
<script>
  $(function() {
    // The rest of my code goes here
  });
</script>

<script>
  jQuery(document).ready(function($) {
    // The rest of my code goes here
  });
</script>
```

- ☒ The code inside them can manipulate the DOM safely, knowing the browser has loaded it fully.
- ☐ The code inside them can manipulate images on the page safely, knowing they have fully downloaded to the browser.
- ☐ The code inside them will be run exactly once per user session.
- ☐ The code inside them is not aware of the DOM.

**Q49. Which describes how jQuery makes working with the DOM faster?**

- ☐ jQuery optimizes the DOM in a background thread, making updates faster.
- ☐ jQuery avoids using the DOM at all.
- ☐ jQuery uses a virtual DOM that batches updates, making inserts and deletes faster.
- ☒ jQuery code to perform DOM manipulation is shorter and easier to write, but does not make DOM operations faster.

**Q50. There are some issues with this snippet of jQuery. First, it is manipulating CSS directly, rather than manipulating classes and leaving the CSS in stylesheets. What else in this code is best to avoid?**

```
$('.item').css('background-color', 'red');
// some other code here
```

```
var firstSubItem = $('.item').find('.sub-item').get(0);
// some other code here too
$('.item').parents('.navigation').css('font-weight', 'bold');
```

- ☐ The `.css()` method accepts only an object, not two separate arguments. This will trigger an exception that should be caught.
- ☐ The `$('.item')` selection is being made several times. This should be cached in a variable for (however slightly) better performance and easier maintainability.
- ☒ The call to `.parents()` is in an inefficient place.
- ☐ All the calls to `$('.item')` should be chained together as a single executable line for better performance.

#### Q51. Which choice is an example of statement chaining?

- ☐ `var $p = $('p'); console.log($p.length);`
- ☒ `$('p').find('a').children('li');`
- ☐ `$('p > a > li');`
- ☐ `$('p'); $('a'); $('li');`

#### Q52. How can you ensure that some code executes only when a class `active` appears on an element?

- ☐ `$('.element').attr('class', 'active')`
- ☐ `$('.element').with('.active')`
- ☒ `$('.element').hasClass('active')`
- ☐ `$('.active').then()`

#### Q53. jQuery has a main function for handling AJAX, and several shorthand function including `load()` that make calling that main function easier. Given this HTML snippet, how can you insert only the second snippet from the `source.html` file (`div#one`) into the `#load-me` div on-demand via AJAX?

```
<div id="load-me">This area will be replaced with AJAX loaded content.</div>
```

```
<div id="one">
```

```
  <h1>First Piece</h1>
```

```
  <p>Lorem ipsum dui maximus quam condimentum dolor eleifend scelerisque.</p>
```

```
</div>
```

```
<div id="two">
```

```
  <h1>Second Piece</h1>
```

```
  <p>Lorem ipsum proin facilisis augue in risus interdum ornare.</p>
```

```
</div>
```

- ☐ \$('#load-me').get('source.html#one');
- ☐ \$('#load-me').get('source.html #one');
- ☒ \$('#load-me').load('source.html #one');
- ☐ \$('#load-me').load('source.html', '#one');

**Q54. Given this HTML list and subsequent two lines of jQuery, what is the difference in the behavior of .closest() and .parents()?**

```
<ul class="items" id="main-menu">
  <li>
    Item 1
    <ul id="sub-menu">
      <li class="leaf">Sub Item 1</li>
      <li>Sub Item 2</li>
    </ul>
  </li>
</ul>
```

```
$('.leaf').closest('.items');
$('.leaf').parents('.items');
```

- ☐ .closest() returns .leaf and #main-menu; .parents() returns #main-menu and #sub-menu.
- ☐ .closest() returns .leaf and #sub-menu; .parents() returns #main-menu and #sub-menu.
- ☐ .closest() returns only #main-menu; .parents() returns #main-menu and #sub-menu.
- ☐ .closest() returns only #sub-menu; .parents() returns #main-menu and #sub-menu.

Source: [jQuery closest Method](#)

**Explanation:** Considering current HTML code, .closest() returns only #main-menu; .parents() returns only #main-menu; cause both of them are looking for .items class which only exist in the #main-menu. Thus all choices are incorrect.

**Q55. What does this line of code do?**

```
$('#ul > li:first-child');
```

- ☒ selects the first list item inside all unordered lists on the page
- ☐ selects the first list item inside the first unordered list on the page
- ☐ selects the first element inside any list items on the page
- ☐ creates a predefined CSS selector that can be reused later

**Q56. Below is a list of items that you want to be clickable and an event handler function. How can you assign the event handler to every item in the list in a**

way that is most performant, and also that ensures that the handler is called even if more items are added to the list programmatically?

```
<ul class="clickable-list">
  <li>First Item</li>
  <li>Second Item</li>
  <li>Third Item</li>
  <li>Fourth Item</li>
  <li>Fifth Item</li>
</ul>

function listResponder(evt) {
  console.log('You clicked a list item that says', evt.target.innerText);
}
```

- ☐ `$('.clickable-list').click(listResponder);`
- ☒ `$('.clickable-list').on('click', 'li', listResponder);`
- ☐ `$('.clickable-list').on('click', append, listResponder);`
- ☐ `$('.clickable-list').each(function() { $(this).click(listResponder); });`

**Q57. What is the difference between `('p').find('a')` and `('p').children('a')`?**

- ☐ `find()` traverses only one level down, whereas `children()` selects anything inside the original element
- ☐ `$('.p').find('a')` finds all paragraphs inside links, whereas `$('.p').children('a')` finds links within paragraph tags
- ☐ `.find()` always searches the entire DOM tree, regardless of the original selection. `.children()` searches only the immediate children of an element
- ☒ `children()` traverses only one level down, whereas `find()` selects anything inside the original element

Source: <https://api.jquery.com/find/>

**Explanation:** Given a jQuery object that represents a set of DOM elements, the `.find()` method allows us to search through the descendants of these elements in the DOM tree and construct a new jQuery object from the matching elements. The `.find()` and `.children()` methods are similar, except that the latter only travels a single level down the DOM tree.

**Q58. Consider the following markup, used to lay out three balls on the page, all hidden. How can you select the green ball, make it faded in over the course of three seconds, and log a console message when the animation has finished?**

```
<div class="balls">
  <div class="ball ball--red" style="display: none"></div>
  <div class="ball ball--green" style="display: none"></div>
  <div class="ball ball--blue" style="display: none"></div>
</div>
```

- ☒ A

```
$('.ball--green').fadeIn(3000, function(){  
    console.log("Animation is done!");  
});
```

- ☐ B

```
$('.ball--green').fade('in',3000).done(function(){  
    console.log("Animation is done!");  
});
```

- ☐ C

```
$('.ball--green').fadeIn(3).console().log("Animation is done!");
```

- ☐ D

```
$('.ball--green').fadeIn("3s", function(){  
    console.log("Animation is done!");  
});
```

Source: [jQuery Docs: fadeIn](#)

Durations are given in milliseconds; higher values indicate slower animations, not faster ones. The strings 'fast' and 'slow' can be supplied to indicate durations of 200 and 600 milliseconds, respectively. If any other string is supplied, or if the duration parameter is omitted, the default duration of 400 milliseconds is used.

### Q59. Why might you use custom events instead of shared helper functions? For example

```
$(document).on('myCustomEvent', function(){  
    // act on my custom event  
});
```

//and later...

```
$(document).trigger('myCustomEvent');
```

- ☐ Custom events are at least an order of magnitude faster than helper functions
- ☒ Custom events can be listened for and acted upon across one or more scripts without needing to keep helper functions in scope
- ☐ Handler functions for custom events are less likely to be mangled by minification and obfuscation build tools
- ☐ It is easier to write documentation for custom events than it is for helper functions

Source: [learn.jquery.com](#)



Instead of focusing on the element that triggers an action, custom events put the spotlight on the element being acted upon. This brings a bevy of benefits, including: Behaviors of the target element can easily be triggered by different elements using the same code. Behaviors can be triggered across multiple, similar, target elements at once. Behaviors are more clearly associated with the target element in code, making code easier to read and maintain.

**Q60. In the HTML and JavaScript below, the animations will all fire at once. How can you make them fire in sequence instead?**

```
<div id="element-1" class="animel"></div>
<div id="element-2" class="animel"></div>
<div id="element-3" class="animel"></div>
```

```
$('#element-1').animate({ top: '+=100' }); $('#element-2').animate({ top: '+=100'
});
$('#element-3').animate({ top: '+=100' });
```

- ☐ A

```
$('#element-1').animate({ top: '+=100' })
  .pushStack('#element-2')
  .animate({ top: '+=100' })
  .pushStack('#element-3').animate({ top: '+=100' })
```

- ☒ B

```
$('#element-1').animate({ top: '+=100' }, function() {
  $('#element-2').animate({ top: '+=100' }, function() {
    $('#element-3').animate({ top: '+=100' });
  })
});
```

- ☐ C

```
$('#element-1').animate({ top: '+=100' })
  .add('#element-2').animate({ top: '+=100' })
  .add('#element-3').animate({ top: '+=100' })
```

- ☐ D

```
$('#element-1').animate({ top: '+=100' }, {queue: 'custom'});
$('#element-2').animate({ top: '+=100' }, {queue: 'custom'});
$('#element-3').animate({ top: '+=100' }, {queue: 'custom'});
$('#custom').dequeue();
```

[Source: jQuery Docs: animate](#)

the `.animate()` method can take in a function to call once the animation is complete, called once per matched element. Which is called the complete option for the animate method

**Q61. Given this checkbox, how can you determine whether a user has selected or cleared the checkbox?**

```
<input type="checkbox" id="same-address" checked>
```

- ☐ by checking the value of `$('#same-address').val()`
- ☒ by checking the value of `$('#same-address').prop('checked')`
- ☐ by checking the value of `$('#same-address').attr('checked')`
- ☐ by checking the value of `$('#same-address').checked`

**Q62. In some projects, jQuery is not included as a file with an obvious version number (if it has been run through a minifier or other code bundler, for example). How can you detect programmatically what version of jQuery is active?**

- ☐ `jQuery.version()`
- ☐ `jQuery.jquery`
- ☐ `jQuery.prototype.version`
- ☒ `jQuery.fn.jquery`

**Q63. Given this snippet of HTML, how can you get the value of the text field using jQuery?**

```
<input type="text" class="form-control" id="firstName" placeholder="" value="" required="">
```

- ☐ `$('input[type=text]').val()`
- ☐ `$('.form-control').val()`
- ☒ all of these answers
- ☐ `$('#firstName').val()`

all the listed selectors will target the text field since it has a `type=text`, a `class=form-control`, and an `id=firstName`

**Q64. Which property of the jQuery event object references the DOM object that dispatched an event?**

- ☒ `target`
- ☐ `self`
- ☐ `source`
- ☐ `object`

[Source: jQuery Docs: event.target](#)

The target property can be the element that registered for the event or a descendant of it. It is often useful to compare event.target to this in order to determine if the event is being handled due to event bubbling.

**Q65. You want to write a plugin that creates a new traversal function—such as parent() and children()—and behaves like the ones jQuery includes out of the box. It needs to correctly modify the list of selections jQuery tracks internally, build up a list of additional items, and return the merged collection. What do you need to return on the last line of the function in order for this plugin to work correctly?**

```
$.fn.myTraverse = function() {  
    // ... setup  
  
    var additionalItems = [ /* some additional items for jQuery */ ];  
  
    return // return what?  
}
```

- ☐ return this.append(additionalItems);
- ☐ return additionalItems.appendTo(this);
- ☒ return this.pushStack(additionalItems);
- ☐ return this.add(additionalItems);

[Source: jQuery Docs](#)

When you call pushStack() off of the current collection, it will take the given collection and associate it to the current collection such that calling the end() method (after the plugin exits) will return the programmer to the current collection.

[Extra Reading: bennadel.com](#)

**Q66. Given this snippet of HTML and jQuery code, what will the result look like?**

```
<ul class="items">  
  <li class="active">Item 1</li>  
  <li>Item 2</li>  
  <li>  
    Item 3  
    <ul>  
      <li>Sub Item 1</li>  
      <li>Sub Item 2</li>  
    </ul>  
  </li>  
</ul>  
  
$('.items').find('.active').nextAll().addClass('after-active');
```

- ☒ A

```
<ul class="items">
  <li class="active">Item 1</li>
  <li class="after-active">Item 2</li>
  <li class="after-active">
    Item 3
    <ul>
      <li>Sub Item 1</li>
      <li>Sub Item 2</li>
    </ul>
  </li>
</ul>
```

• ☐ B

```
<ul class="items">
  <li class="active">Item 1</li>
  <li class="after-active">Item 2</li>
  <li class="after-active">
    Item 3
    <ul class="after-active">
      <li>Sub Item 1</li>
      <li>Sub Item 2</li>
    </ul>
  </li>
</ul>
```

• ☐ C

```
<ul class="items">
  <li class="active">Item 1</li>
  <li class="after-active">Item 2</li>
  <li class="after-active">
    Item 3
    <ul>
      <li class="after-active">Sub Item 1</li>
      <li class="after-active">Sub Item 2</li>
    </ul>
  </li>
</ul>
```

• ☐ D

```
<ul class="items">
  <li class="active">Item 1</li>
  <li class="after-active">Item 2</li>
  <li class="after-active">
    Item 3
    <ul class="after-active">
      <li class="after-active">Sub Item 1</li>
      <li class="after-active">Sub Item 2</li>
    </ul>
  </li>
</ul>
```

Source: [jQuery Docs](#)

**.nextAll([selector]) method** Gets all following siblings of each element in the set of matched elements, optionally filtered by a selector.

**Q67. You have an element with a series of code (not CSS) animations applied to it that could be triggered by code you control, or other code elsewhere (such as plugins). How can you fire some code when all those animations have completed?**

- ☐ A

```
$('#element').on('animationend', function () {  
    console.log('Finally, everything is done!');  
});
```

- ☐ B

```
$('#element')  
    .on('promise')  
    .then(function () {  
        console.log('Finally, everything is done!');  
    });
```

- ☐ C

```
$('#element')  
    .promise()  
    .catch(function () {  
        console.log('Finally, everything is done!');  
    });
```

- ☐ D

```
$('#element')  
    .promise()  
    .then(function () {  
        console.log('Finally, everything is done!');  
    });
```

1. [Source: HTMLElement: animationend event | MDN](#)
2. [Example: Stackoverflow](#)

**Q68. HTML5 data attributes allow you to create valid custom attributes to store arbitrary data within DOM elements. jQuery has an API to interface with custom data such as the series of quotes below. How can you mark the second quote as your favorite?**

```
<div class="quotes">  
    <blockquote data-favorite="false">A quote</blockquote>  
    <blockquote data-favorite="false">A favorite quote</blockquote>  
    <blockquote data-favorite="false">A quote</blockquote>
```

```
<blockquote data-favorite="false">A quote</blockquote>
</div>
```

- ☐ `$('.blockquote').second().attr('favorite', true);`
- ☒ `$('.blockquote:nth-child(2)').data('favorite', true);`
- ☐ `$('.blockquote').second().data('favorite', true);`
- ☐ `$('.blockquote:nth-child(2)').attr('favorite', true);`

1. [Source: .data\(\) | jQuery API Documentation](#)
2. [Source: :nth-child\(\) | MDN Docs](#)

**Q69. jQuery can create event handlers that execute exactly once. How is this done?**

- ☐ `$('.button').click(function() { console.log('this will only happen once'); }, false);`
- ☐ `$('.button').on('click', function() { console.log('this will only happen once'); }).off('click');`
- ☐ `$('.button').once('click', function() { console.log('this will only happen once'); });`
- ☒ `$('.button').one('click', function() { console.log('this will only happen once'); });`

[Source: .one\(\) | jQuery API Documentation](#)

**Q70. You want to implement the behavior of an effect like `slideDown()` manually using `animate()`. What is one critical point you need to remember?**

- ☐ `slideDown()` requires animating the background color; doing so with `animate()` requires the jQuery Color plugin.
- ☒ `slideDown()` includes toggling visibility automatically. `animate()` does not automatically set any properties.
- ☐ `slideDown()` requires the element to have a height set in pixels. `animate()` does not.
- ☐ Effects created with `animate()` must be run over at least 100 milliseconds, where `slideDown()` can run as quickly as 50ms.

[Source: .slideDown\(\) | jQuery API Documentation](#)

[Source: .animate\(\) | jQuery API Documentation](#)

**Q71. What is the main difference between the `contents()` and `children()` functions?**

- ☐ They both return the content of selected nodes, but `children()` also includes text and comment nodes.
- ☐ The `contents()` function only includes text nodes of the selected elements.
- ☐ The `children()` function only includes text nodes of the selected elements.
- ☒ They both return the content of selected nodes, but `contents()` also includes text and comment nodes.

1. [Source: `.children\(\)` | jQuery API Documentation](#)
2. [Source: `.contents\(\)` | jQuery API Documentation](#)

**Q72. If your JavaScript project involves a lot of DOM manipulation, but no AJAX or animation, which version of jQuery should you use?**

- ☐ jQuery 3 compressed
- ☐ jQuery 3 slim
- ☐ jQuery 2
- ☒ None of these - jQuery requires AJAX

**Q73. The `.ready()` function is one of the most basic parts of jQuery, but jQuery also provides a mechanism for executing code when both one or more Promises have resolved and the DOM is ready. Which code snippet accomplishes this?**

- ☐ A

```
$(function({
  getData : $.get('http://httpbin.org/get'),
  delayedData : $.get('http://httpbin.org/delay/3')
})) {
  //DOM is ready, getData and delayedData are available
});
```

- ☐ B

```
$(($.get('http://httpbin.org/get'),
$.get('http://httpbin.org/delay/3')).then(function (
  getData,
  delayedData,
) {
  //DOM is ready, getData and delayedData are available
}));
```

- ☐ C

```
$.when($.get('http://httpbin.org/get'),
$.get('http://httpbin.org/delay/3')).then(function (
  getData,
```

```
    delayedData,  
  ) {  
    //DOM is ready, getData and delayedData are available  
  });
```

- ☒ D

```
$.ready($.get('http://httpbin.org/get'),  
$.get('http://httpbin.org/delay/3')).then(function (  
  getData,  
  delayedData,  
) {  
  //DOM is ready, getData and delayedData are available  
});
```

**Q74. You want to take an element and any event handlers that go with it out of the DOM to do some work—without the changes affecting the rest of the page—and then move it somewhere else in the DOM, like right after the opening tag. What should go on the first line of this code snippet?**

```
// what goes here?  
// ... do some other hidden work on $example  
$example.prependTo(document.body);
```

- ☐ var \$example = \$('#example').remove();
- ☐ var \$example = \$('#example').clone();
- ☒ var \$example = \$('#example').detach();
- ☐ var \$example = \$('#example').addBack().empty();

<https://api.jquery.com/detach/>

**Q75. Review the HTML below. You want to select the first item in the list and fade it out, then select the subsequent items up to (but not including) the active item, and fade them out halfway. How can you set up a single chain to do this?**

```
<ul class="items">  
  <li>Item 1</li>  
  <li>Item 2</li>  
  <li>Item 3</li>  
  <li class="active">Item 4</li>  
  <li>Item 5</li>  
  <li>Item 6</li>  
</ul>
```

- ☒ A

```
$('.items > li').first().fadeOut().nextUntil('.active').fadeTo('fast', 0.5);
```



- ☐ B

```
$('.items').children(':first-child').fadeOut().filter('.active').fadeTo('fast', 0.5);
```

- ☐ C

```
$('.items > li').first().fadeOut().nextAll('.active').fadeOut(50);
```

- ☐ D

```
$('.items').find('li:first-child').fadeOut().next('.active').fadeTo('fast', 0.5);
```

1. <https://api.jquery.com/fadeTo/>
2. <https://api.jquery.com/fadeOut/>
3. <https://api.jquery.com/nextUntil/>

#### Q76. What is a particular performance concern when dealing with event handlers, and how can you cope with it?

- ☐ Finding which element an event occurred on is expensive. Assign most events to document.body and use .is() to act on the element of interest.
- ☒ Some events, such as mousemove and scroll, happen a lot on a typical page. Debounce or throttle their handlers to make sure the handlers are not called more than you actually need.
- ☐ Listening for an event that does not exist can create serious memory leaks. Be careful to spell event names correctly to avoid consuming too much memory.
- ☐ DOM elements with an ID will fire events more efficiently than with classes. Always use IDs instead of classes where possible.

#### Q77. What is the purpose of the jQuery.fx.off global property?

- ☐ It turns off animations that are used to provide motion effect, but appearance effects remain enabled.
- ☐ It causes animation effects that are triggered via functions to instead be executed using CSS.
- ☒ It globally disables all animations. When animations are run, all animation methods will immediately set elements to their final state when called, rather than displaying an effect.
- ☐ It globally disables animations that are triggered by CSS class changes.

Source: [jquery.fx.off](http://jquery.fx.off) Property

**Q78. When you use custom JQuery selection extensions, such as :animated, on a page with lots of DOM elements, you can run into performance issues. What is the best practice for managing those issues?**

- ☐ Select `$(document.body)` first, then use `.filter` with the custom extension.
- ☐ Use the custom extension with `.has()`.
- ☐ Start with the custom extension, then use `.find` with a selector that exists in CSS to limit the selection.
- ☐ Start with a selection that exists in CSS, then refine the selection using `.filter()` with the custom extension.

**Q79. What is the main difference between the `ajaxStop` and `ajaxComplete` global handlers?**

- ☒ `ajaxStop` fires when all open requests have completed, while `ajaxComplete` fires when an individual request completes.
- ☐ There is no difference. They are aliases of each other.
- ☐ `ajaxStop` fires when an error has been encountered, while `ajaxComplete` fires when a successful request completes.
- ☐ `ajaxComplete` fires when all open requests have completed, while `ajaxStop` fires when an individual request completes.

**Q80. The `:only-child` selector selects \_.**

- ☐ all elements that are child elements.
- ☐ elements that have no siblings.
- ☐ elements that have only one child element.
- ☒ only the child elements of a given element.